Study of Throughput and Delay in Finite-Buffer Line Networks

Badri N. Vellambi
Institute for Telecommunications Research
University of South Australia
Mawson Lakes, Australia
E-mail: {badri.vellambi}@unisa.edu.au

Nima Torabkhani, Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250
E-mail: {nima, fekri}@ece.gatech.edu

***Abstract*—In this work, we study the effects of finite buffers on the throughput and delay of line networks with erasure links. We identify the calculation of performance parameters such as throughput and delay to be equivalent to determining the stationary distribution of an irreducible Markov chain. We note that the number of states in the Markov chain grows exponentially in the size of the buffers with the exponent scaling linearly with the number of hops in a line network. We then propose a simplified iterative scheme to approximately identify the steady-state distribution of the chain by decoupling the chain to smaller chains. The approximate solution is then used to understand the effect of buffer sizes on throughput and distribution of packet delay. Further, we classify nodes based on congestion that yields an intelligent scheme for memory allocation using the proposed framework. Finally, by simulations we confirm that our framework yields an accurate prediction of the variation of the throughput and delay distribution.**

I. Introduction

In networks, packets that have to be routed from one node to the other may have to be relayed through a series of intermediate nodes. Also, each node in the network may receive packets via many data streams that are being routed simultaneously from their source nodes to their respective destinations. In such conditions, the packets may have to be stored at intermediate nodes for transmission at a later time. If buffers are unlimited, the intermediate nodes need not have to reject or drop packets that arrive. For practical reasons, buffers are limited in size. Although a large buffer size is preferred to minimize packet drops, large buffers have an adverse effect on the mean and variance in packet delay. Additionally, as second-order effects, using larger buffer sizes at intermediate nodes would have practical problems such as on-chip board space and increased memory-access latency. Though our work is motivated partly by such concerns, our work is far from modeling realistic scenarios. This work modestly aims at providing a theoretical framework to understand the fundamental limits of single information flow in finite-buffer line networks and investigate the trade-offs between throughput, packet delay and buffer size.

The problem of computing capacity[1] and designing efficient coding schemes for lossy wired and wireless networks has been widely studied [1]–[4]. However, the study of capacity of networks with finite buffer sizes has been limited. This can be attributed solely to the fact that the analysis of finite buffer systems are generally more challenging. With the advent of network coding [4], [5] as an elegant and effective tool for attaining optimum network performance, the interest in the study of finite buffer networks has been increased.

The problem of studying throughput and delay of networks with finite buffers has also been studied in queueing theory. These problems can be seen to be similar, since the packets can be viewed as customers and the delay due to packet loss in the link as the arbitrary service time. Also, the phenomenon of packet overflow in the network can be modeled by a type II blocking (commonly known as *blocking after service*) in stochastic networks. However, there is a subtle difference in the packet-customer analogy when the network has nodes that can send packets over multiple paths to the destination. When such is the case, the node can choose to duplicate packets on both the paths, an event that cannot be captured directly in the customer-server based queueing model. However, that is not the case in line networks. Therefore, the problem of finding buffer occupancy distribution and consequently throughput and delay in certain networks is then seen to be identical to determining certain arrival/departure processes in an open stochastic network of a given topology [6]–[9]. Such relevant works in the field of queueing theory consider a continuous-time model for arrival and departure of packets in the network.

In [10], Lun *et al.* consider the discrete-time analogue of the arrival process by lumping time into epochs (wherein each node can transmit and receive a packet) to analyze the capacity of a simple two-hop lossy network. In our previous work [11], we derived bounds on the throughput of line networks, which were unable to provide good approximations for packet delay and buffer occupancy statistics. While our approach employs a model of network similar to that in [10], [11], we extend their results not only to derive estimates for the capacity of line networks of any hop-length and intermediate node buffer size, but also to derive quantitative estimates for packet delay distribution. Our contributions to this area of research is summarized below.

1. We extend a Markov-chain based modeling to present an iterative estimate for the buffer occupancy distribution at intermediate nodes.
2. Using the estimate, we derive expressions for throughput and packet delay distribution that are seen to be fairly accurate in predicting the actual system behavior.

This work is organized as follows. First, we present the formal definition of the problem and the network model in Section II. Next, we introduce our analysis for finite-buffer line networks in Section III and then investigate packet delay in Section IV. We compare our analytical results with actual

[1] Throughout this work, we use capacity to mean the supremum of all rates of information flow achievable by any coding scheme.

simulations in Section V and in Section VI we briefly discuss trade-offs between throughput, delay and memory. Finally, Section VII concludes the paper.

II. Problem Statement and Network Model

As illustrated in Figure 1, a line network is a directed graph of h hops with $V = \{v_0, v_1, \dots, v_h\}$ and $\overrightarrow{E} = \{(v_i, v_{i+1}) : i = 0, \dots, h-1\}$ for some integer $h \geq 2$. In the figure, the intermediate nodes are shown by black ovals. The links are assumed to be unidirectional, memoryless and lossy. We let ε_i denote the packet erasure probability over the link (v_i, v_{i+1}). The erasures model only the quality of links (e.g., presence of noise, interference) and do not represent packet drops due to finite buffers. A lossless hop-by-hop acknowledgement setup is in place to indicate the successful receipt of a packet[2]. Moreover, the packet processes on different links are assumed to be independent. Each node $v_i \in V$ has a buffer of m_i packets with each packet having a fixed size of S bytes. Note that the buffer size can vary with the node index. Lastly, the source and destination nodes are assumed to have sufficient memory to store any amount of data.

The system is analyzed using a discrete-time model, where each node can transmit at most one packet over a link per epoch. We let $\{X_i(l)\}_{\mathbb{Z}_{\geq 0}}$ to be the random process denoting the erasure occurrences on the link (v_{i-1}, v_i) at time l. We set $X_i(l) = 0$ if a packet is erased on (v_{i-1}, v_i) at epoch l and $X_i(l) = 1$ otherwise.

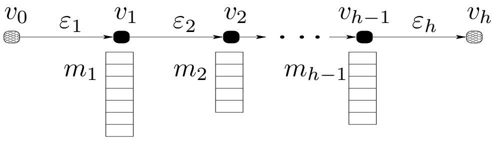

Fig. 1. An illustration of the line network.

The unicast capacity between a pair of nodes is defined to be the supremum of all achievable rates of transmission of information packets (in packets per epoch) between a pair of nodes. The supremum is calculated over all possible means used for packet generation and buffer update at intermediate nodes. Note that the source node can generate innovative packets during each epoch. For instance, in the particular case of the line network of Figure 1, we would like to identify the unicast capacity between v_0 and v_h.

Before we proceed to the modeling, we briefly motivate the assumed discrete-time model with an example. Consider a continuous-time model with the discrete-time model for varying times of epoch for a simple continuous-time two-hop line network with a Poisson packet generation process at the source with parameter $\lambda_1 = 10$ pkts/sec. The service time at the intermediate node is also Poisson with parameter $\lambda_2 = 10$ pkts/sec, and the links connecting the source to the intermediate node and the intermediate node to the destination are both packet-erasure channels with erasure probabilities $\varepsilon_1 = \varepsilon_2 = 0.1$. Finally, suppose that the intermediate node has a finite buffer of $m = 10$ packets. Figure 2 presents the (simulated) capacity for the continuous model and the time-discretized models for various epoch durations. It is noticed that as the epoch duration is made smaller, the discrete-time model becomes more accurate in predicting the capacity. This was verified to be the case for all line networks with Poisson arrivals and service times.

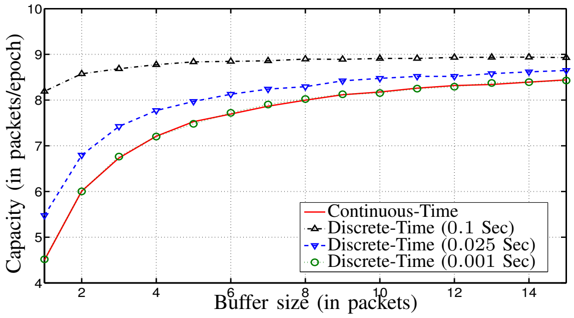

Fig. 2. An illustration of the precision of the discrete-time model.

Lastly, we use the following notations. $\mathbb{G}(t)$ denotes the geometric distribution with mean inter-arrival time $t \in \mathbb{R}$. $\sigma(\cdot)$ denotes the indicator function for $\mathbb{Z}_{>0}$. For any $x \in \mathbb{R}$, $\overline{x} \triangleq 1 - x$. Lastly, $\otimes$ denotes the convolution operator.

III. Finite-Buffer Analysis

Here, we develop our framework for analyzing finite-buffer erasure networks. In the sections that follow, these techniques will be used to study various the performance indicators. We proceed as follows.

1) Rate-optimal Schemes: One of the most important performance parameters of a network is its throughput and the problem of identifying capacity is directly related to the problem of finding schemes that are *rate-optimal*. In our model of line network, a scheme that performs the following in the same order can be seen to be rate-optimal.

1. If the buffer of a node is not empty at a particular epoch, then it must transmit at least one of the packets.
2. A node deletes the packet transmitted at an epoch if it receives an acknowledgement from the next hop.
3. A node accepts an arriving packet if it has space in its buffer. It then sends an ACK to the previous node.

In the absence of feedback, rate-optimality can be achieved by employing random linear combinations based network coding over a large finite field as is described in [10], [11].

2) Markov Chain Modeling of the Buffer States at Intermediate Nodes: In order to model the network with lossless feedback, we need to track the number of packets that each node possesses at every instant of time. We do so by using the rules of buffer update under the optimal scheme. Let $\mathbf{n}(l) = (n_1(l), \dots, n_{h-1}(l))$ be the vector whose i^{th} component denotes the number of packets the i^{th} intermediate node possesses at time l. The variation of state at the l^{th} can be tracked using auxiliary random variables $Y_i(l)$ defined by

$$Y_i(l) = \begin{cases} \sigma(n_{i-1}(l)) X_i(l) & i = h \\ X_i(l) \sigma(m_i - n_i(l) + Y_{i+1}(l)) & i = 1 \\ \sigma(n_{i-1}(l)) X_i(l) \sigma(m_i - n_i(l) + Y_{i+1}(l)) & 1 < i < h \end{cases}$$

[2] This assumption is made to simplify modeling. In the absence of perfect ACK, one can use random linear coding over a large finite field to achieve the same desired throughput.

The above definition is such that $Y_i(l) = 1$ only if all the following conditions are met.

1. Node v_{i-1} has a packet to transmit to v_i.
2. The link (v_{i-1}, v_i) does not erase the packet at the l^{th} epoch, *i.e.*, $X_i(l) = 1$.
3. Node v_i has space after it has updated its buffer for any changes due to its transmission at that epoch.

The dynamics of buffer states can then by seen to be

$$n_i(l+1) = n_i(l) + Y_i(l) - Y_{i+1}(l), \qquad i = 1, \ldots, h-1. \quad (1)$$

Note that since $\mathbf{Y}(l) = (Y_1(l), \ldots, Y_h(l))$ is a function of $\mathbf{n}(l)$ and $\mathbf{X}(l) = (X_1(l), \ldots, X_h(l))$, $\mathbf{n}(l+1)$ depends only on its previous state $\mathbf{n}(l)$ and the channel conditions $\mathbf{X}(l)$ at the l^{th} epoch. Hence, we see that $\{\mathbf{n}(l)\}_{l \in \mathbb{Z}_{\geq 0}}$ forms a Markov chain. It is readily checked that this chain has $\prod_{i=1}^{h-1}(m_i + 1)$ states. Further, this chain is *irreducible, aperiodic, positive-recurrent*, and *ergodic* [12] and therefore has a unique steady-state probability p^∞. By ergodicity, we can obtain temporal averages by statistical averages. We then see that the computation of (throughput) capacity is equivalent to the computation of the likelihood of the event that $Y_h = 1$.

3) Approximated MC for an intermediate node: The exponential growth in the size of the chain and the presence of boundaries (due to finite buffers), exact calculation of the steady-state probabilities (and hence the throughput) becomes very cumbersome even for networks of reasonable buffer sizes and hop-lengths. The exact chain for the dynamics of the system is such that a state update at a node has a strong dependence on the states of both its previous-hop and its next-hop neighbors. Additionally, the process of packet transmission over intermediate edge can be shown to be non-memoryless. These facts add to the intractability of the exact computation of the distribution. However, it is possible to decouple the chain into several Markov chains with a single finite-boundary under some simplifying assumptions. To have an approximate decoupled model, we need to identify the transition probabilities of the decoupled chains, which is possible only if we know the arrival and departure processes on each edge. The rate of information on any edge is directly related to the fraction of time the sending node is non-empty and the fraction of time a successfully delivered packet will get blocked (and this happens if the receiving node is full at the time of packet arrival). Hence, to have a model for a node, we need to have the approximate buffer occupancy distributions for neighboring nodes. This hints naturally at an *iterative* approach to the problem. In this section, we develop an iterative estimation method that considers the effect of blocking with some simplifying assumptions. To develop an iterative technique, we assume the following.

A1. The packets are ejected from nodes in a memoryless fashion. Equivalently, we assume that $\Pr[(n_{i-1}(t) > 0) \wedge (X_i(t) = 1) | n_i(t) = k]$ does not vary with the occupancy k of the i^{th} node. This allows us to track just the information rate and not the exact statistics.

A2. The blocking event occurs independent of the state of a node, *i.e.*, $\Pr[(Y_{i+1}(l) = 0) \wedge (X_{i+1}(l) = 1) | (n_i(t) = k)]$ is the same for $k = 1, \ldots, m_i$. This allows us to track just the blocking probability and not the joint statistics.

A3. At any epoch, given the occupancy of a particular node, the arrival process is independent of the blocking process.

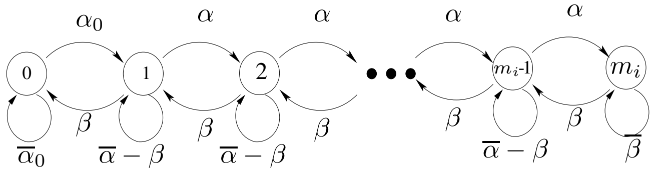

Fig. 3. The chain for the node v_i obtained by the assumptions A1-A3.

These assumptions spread the effect of blocking equally over all non-zero states of occupancy at each node. Under these assumptions, when we are given that the arrival rate of packets as seen by v_i is r_i packets/epoch, and that the probability of blocking by v_{i+1} is $p_{b_{i+1}}$, the dynamics of the state change for the node v_i is given by the chain in Figure 3 with the parameters set to

$$\begin{array}{rcl} \alpha & = & r_i(\varepsilon_{i+1} + \overline{\varepsilon}_{i+1} p_{b_{i+1}}) \\ \beta & = & (1 - r_i)\overline{p_b}_{i+1}\overline{\varepsilon}_{i+1} \\ \alpha_0 & = & r_i \end{array} . \quad (2)$$

For this chain in Fig. 3, the steady-state distribution can be computed to be

$$\Pr[n_i = k] = \varphi(k | r_i, \varepsilon_{i+1}, p_{b_{i+1}}) \triangleq \begin{cases} \frac{1}{1 + \frac{\alpha_0}{\beta}(\sum_{l=0}^{m_i - 1} \frac{\alpha^l}{\beta^l})} & k = 0 \\ \frac{\frac{\alpha_0 \alpha^{k-1}}{\beta^k}}{1 + \frac{\alpha_0}{\beta}(\sum_{l=0}^{m_i - 1} \frac{\alpha^l}{\beta^l})} & k \neq 0 \end{cases}$$

The blocking probability that the node v_{i-1} perceives from the node v_i, assuming that v_i sees an arrival rate of r_i from v_{i-1} and a blocking probability of $p_{b_{i+1}}$ caused by v_{i+1}[3], can then be calculated as follows.

$$p_{b_i} = \begin{cases} (\varepsilon_{i+1} + \overline{\varepsilon}_{i+1} p_{b_{i+1}}) \varphi(m_i | r_i, \varepsilon_{i+1}, p_{b_{i+1}}) & i < h-1 \\ \varepsilon_{i+1} \varphi(m_i | r_i, \varepsilon_{i+1}, 0) & i = h-1 \end{cases} \quad (3)$$

Similarly, the arrival rate at on each edge can be seen to be related to the occupancy of the previous node and the channel erasure probability in the following fashion.

$$r_{i+1} = \begin{cases} \overline{\varepsilon}_i(1 - \varphi(0 | r_i, \varepsilon_{i+1}, p_{b_{i+1}})) & 1 < i < h-1 \\ \overline{\varepsilon}_i(1 - \varphi(0 | r_i, \varepsilon_{i+1}, 0)) & i = h-1 \end{cases} \quad (4)$$

Given two vectors $\mathbf{r} = (r_1, \ldots, r_h) \in [0,1]^h$ and $\mathbf{p_b} = (p_{b_1}, \ldots, p_{b_h}) \in [0,1]^h$, we term $(\mathbf{r}, \mathbf{p_b})$ as an approximate solution to the chain, if they satisfy the equations (3), and (4) in addition to having $r_1 = \overline{\varepsilon}_1$ and $p_{b_h} = 0$. The following theorem guarantees both the uniqueness and the method of identifying the approximate solution to the chain.

Theorem 1: Given a line network with link erasures $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_h)$ and intermediate node buffer sizes $\mathcal{M} = (m_1, \ldots, m_{h-1})$, there is exactly one approximate solution $(\mathbf{r}^*(\mathcal{E}, \mathcal{M}), \mathbf{p_b}^*(\mathcal{E}, \mathcal{M}))$ to the chain. Moreover, this solution can be found iteratively.

While the proof of uniqueness is omitted for brevity, the method of identifying the approximate solution follows a

[3] Note that the arrival rate at the node v_1 is $r_1 = \overline{\varepsilon}_1$ and that the blocking probability of v_h is zero.

simple iterative scheme. First, $\mathbf{p_b}$ is set to $\mathbf{0}$ and 4 is invoked to estimate $\mathbf{r}$. This estimate is then used in 3 to update $\mathbf{p_b}$, which is then used to find the next estimate of $\mathbf{r}$. These iterations repeated until the vectors converge. We omit the details of proof of convergence for lack of space.

Finally, the unicast capacity $\mathcal{C}^*(\mathcal{E}, \mathcal{M})$ can be estimated using the approximate solution from the following.

Corollary 1: The approx. solution $(\mathbf{r}^*(\mathcal{E}, \mathcal{M}), \mathbf{p_b}^*(\mathcal{E}, \mathcal{M}))$ satisfies the flow conservation relations

$$\mathcal{C}^*(\mathcal{E}, \mathcal{M}) = r_1^*(1 - p_{b_1}^*) = r_i^*(1 - p_{b_i}^*), \ i = 1, \ldots, h.$$

IV. Packet Delay Distribution

In this section, we use the approximate solution of Section III-3 to obtain the estimates on the probability distribution of the delay of an information packet. We define the packet delay as the time taken from the instant when the source starts sending the packet to the instant when the destination receives it. In addition to the discussion in section III-1, we assume a *first-come first-serve* treatment of packets at the intermediate node buffers.

In order to compute the distribution of delay that a packet experiences in the network, one can proceed in a hop-by-hop fashion. Considering the last relay node, the additional delay of an arriving packet (at time l) at node v_{h-1} depends on the occupancy of the node v_{h-1} and the erasure channel that follows it to the destination. Suppose at epoch l, node v_{h-1} has $k \leq m_{h-1} - 1$ packets in addition to the arriving packet. Then, the packet has to wait for the first k packets to leave before it can be serviced. Since each transmission takes place independently, the distribution of delay is sum of $k+1$ independent geometric distribution with mean inter-arrival time $\frac{1}{1-\varepsilon_h}$, which is denoted by $\otimes^{k+1}\mathbb{G}(\frac{1}{1-\varepsilon_h})$. Suppose that the distribution of buffer occupancy *at time of packet arrival* is given by $\pi_{h-1}(i)$, then the distribution of delay added by v_{h-1} to the packet is

$$\mathbf{D}_{h-1} = \sum_{i=0}^{m_{h-1}-1} \pi_{h-1}(i) \otimes^{i+1} \mathbb{G}((1 - \varepsilon_h^{\prime})^{-1}). \tag{5}$$

However, the situation is different for other intermediate delays because of the effect of blocking. The additional delay incurred while being stored at the node v_j, $0 < j < h - 1$, is given by

$$\mathbf{D}_j = \sum_{i=0}^{m_j-1} \pi_j(i) \otimes^{i+1} \mathbb{G}((1 - \varepsilon_{j+1}^{\prime})^{-1}), \tag{6}$$

where we used the following to consider blocking.

$$\varepsilon_i^{\prime} = \begin{cases} \varepsilon_i + \theta_{v_i}(m_i)(1 - \varepsilon_i) & i = 1, 2, \ldots, h-1 \\ \varepsilon_h & i = h \end{cases}, \tag{7}$$

where $\theta_{v_i}(k)$ is the steady state probability of that node v_i already has k packets when the packet is transmitted successfully from v_{i-1}. $\pi_j(i)$ and $\theta_{v_i}(k)$ are related by

$$\pi_j(i) = \begin{cases} \frac{\theta_{v_j}(i)}{1-\theta_{v_j}(m_j)} & i = 1, 2, \ldots, m_j - 1 \\ 0 & i = m_j \end{cases}. \tag{8}$$

By assuming that the delays incurred by each node and its adjoining outgoing link is independent of each other, we obtain the total delay considering all hops to be

$$\mathbf{D} = \mathbb{G}((1 - \varepsilon_1^{\prime})^{-1}) \otimes \mathbf{D}_1 \otimes \cdots \otimes \mathbf{D}_{h-1}. \tag{9}$$

Hence, the delay distribution is known if the steady-state distributions of buffer states $(\pi_j(\cdot), \ j = 1, ..., h-1)$ as seen by arriving packets is known. However, it is a simple exercise to derive these distributions from the results of Section III-3.

V. Results on Finite-Buffer Analysis

We have so far presented some fundamental tools for finite-buffer analysis of line networks. In this section, we show that they are very helpful to obtain accurate estimates of the performance parameters such as throughput, delay distribution and buffer occupancy distribution for line networks.

To understand the variation of our capacity estimate of Section III, in each of the figures, the simulation of the actual capacity is presented in addition to our analytical results. Figure 4 presents the variation of the capacity with the hop length for a network with each intermediate node having a buffer size of five packets. Moreover, the simulations are performed when the probability of erasure on every link is set to either 0.25 or 0.5. It is noticed that the estimate captures the variation of the actual capacity of the network within about 1.5% of error.

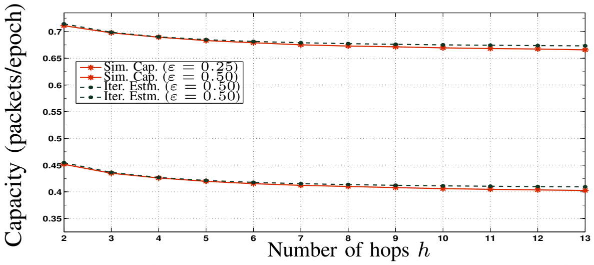

Fig. 4. Capacity of a line network with $m = 5$ vs. the number of hops h.

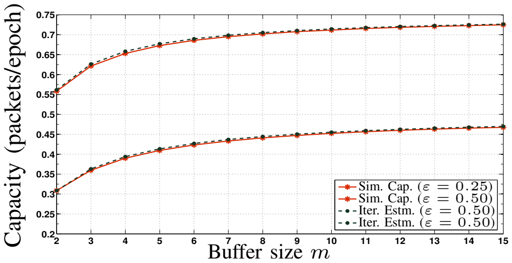

Fig. 5. Capacity of a line network with $h = 8$ vs. the buffer size m.

In order to study the effect of buffer size, we simulated a line network of eight hops having the same erasures as in the previous setting. Figure 5 presents the variation of our results and the actual capacity as the buffer size of the intermediate node is varied. It can be seen that as the buffer size is increased, all curves approach the ideal min-cut capacity of $1 - \varepsilon$.

Figure 6 presents the variation of delay distribution with respect to the buffer size for an eight-hop line network with the erasure probability on every link set to 0.25. It can be seen that both the mean and the variance of the distribution increases with the increase in the buffer size. It is noted that the analytic prediction of the delay is more conservative than the actual simulation i.e., the analytic estimate of the variance is higher than the actual simulated one.

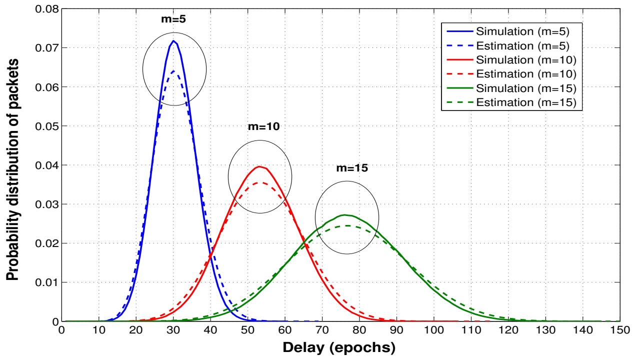

Fig. 6. Delay distribution in an 8 hop line network for varying buffer sizes.

VI. Throughput and Delay trade-offs

Based on our the computed buffer occupancy distributions, we categorize the nodes into 3 types according to their buffer occupancy. Nodes are of Type 1, 2 or 3 depending on whether the rate of incoming rate of innovative packets is larger, same, or smaller, respectively, than the possible rate of innovative packets that can be sent on the outgoing edge. The typical occupancy distributions of such nodes are presented in Figure 7. The figure presents a classification of nodes from a four-hop line network with $\varepsilon_1 = 0.2, \varepsilon_2 = 0.5, \varepsilon_3 = 0.5, \varepsilon_4 = 0.2$ when the three intermediate nodes have buffer sizes $m = 10$ and $m = 30$, respectively. Note that in this example, Node v_i is of Type i. While Type 3 nodes are generally starved and Type 1 nodes are generally full, Type 2 nodes have a near uniform distribution. Increasing the buffers of Type 1 or Type 3 nodes does not affect their blocking probabilities or the general shape of the occupancy distribution. However, increasing the buffer sizes of Type 2 nodes decreases the blocking probability of such nodes. Note that while the classification of these nodes is a trivial exercise, the means to identify/estimate the arrival and departure rates is non-trivial.

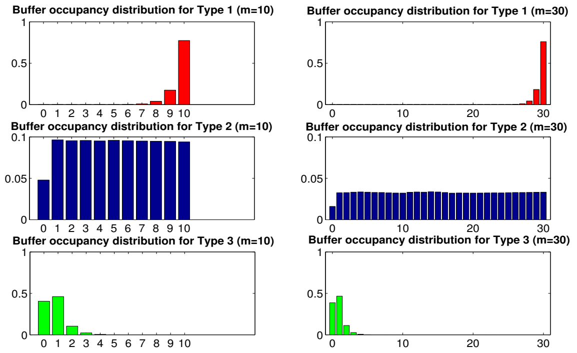

Fig. 7. Buffer occupancy distribution for different types of nodes

Figure 8 shows the effect of increasing buffer size of nodes of a particular type on the throughput and delay in the mentioned line network example. In order to observe the effect of the chosen node of a particular type, the buffers of nodes of other types are fixed at five packets and buffer size of the desired node is varied. We conclude that by allocating memory to Type 1 nodes yields minimal improvement in throughput and a large increase in delay. However, the effect of increasing the buffers of Type 3 nodes has almost no effect on both parameters. However, allocating memory to Type 2 nodes can affect the throughput sufficiently although with a moderate increase in the expected delay.

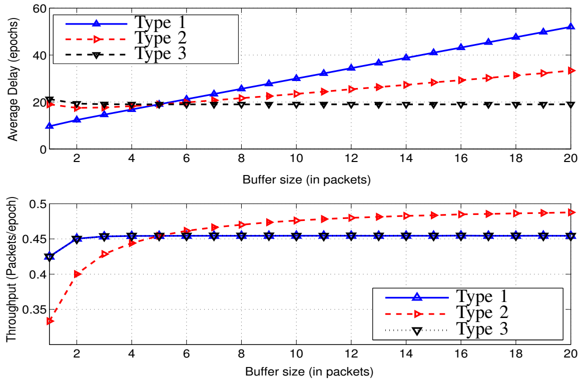

Fig. 8. Throughput Vs. Average Delay for different type of nodes

VII. Conclusions

We presented an approximate markov-chain based model for analyzing the dynamics of finite-buffer line networks. The model provides an iterative procedure for computing the distribution of buffer occupancy as a step in the estimation of throughput capacity of such networks. The model was seen to provide an accurate computation of throughput for varying buffer sizes, hop-length and channel erasure probabilities. The computed buffer occupancy distribution was then used to study the distribution of packet delay in such networks. This proposed model was used to identify the level of congestion in intermediate nodes, yielding a rule for intelligent memory allocation in such networks. The proposed scheme was seen to near-precisely track the dynamics and variations of the investigated performance metrics.

References

[1] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks.," *IEEE Trans. on Information Theory*, vol. 52, no. 3, pp. 789–804, 2006.

[2] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," *IEEE Intl. Symposium on Inform. Theory*, Sept. 2005.

[3] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, 2003.

[4] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. on Inform. Theory*, vol. 49, pp. 371–381, February 2003.

[5] D. S. Lun, M. Medard, and M. Effros, "On coding for reliable communication over packet networks," *in Proc. of Allerton conference on communication, control, and computing, Monticello, IL, 2004*.

[6] T. Altiok, "Approximate analysis of queues in series with phase-type service times and blocking," *Oper. Res.*, vol. 37, pp. 301–310, July 1989.

[7] A. Brandwajn and Y.-L. L. Jow, "An approximation method for tandem queues with blocking," *Oper. Res.*, vol. 36, no. 1, pp. 73–83, 1988.

[8] K. C. So and K. tsai E. Chin, "Performance bounds on multiserver exponential tandem queues with finite buffers," *European Journal of Operational Research*, vol. 63, no. 3, pp. 463–477, 1992.

[9] R. Serfozo, *Introduction to Stochastic Networks*. New York: Springer-Verlag, 1st ed., 1999.

[10] D. S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter, "An analysis of finite-memory random linear coding on packet streams," *in Proc. of the 2nd Workshop on Network Coding, Theory, and Applications (NetCod 2006), Boston, MA, April 3-7, 2006*.

[11] B. N. Vellambi, N. Rahnavard, and F. Fekri, "The effect of finite memory on throughput of wireline packet networks," *in the Proc. of Information Theory Workshop (ITW 2007), Lake Tahoe, CA*, Sept. 2007.

[12] J. R. Norris, *Markov Chains*. Cambridge University Press, 1998.